Coursera and Campuswire Query Link

CS410 Project

Xiaofan Liu / Boyu Pang / Brittany West

December 2021

Introduction

Project Process

- Collect data from Coursera
- Cleaning the dataset
- Build the data corpus
- Collect sample queries from Campuswire
- Generating query result judgement manually
- Ranking function implementation
- Testing, fine-tuning and improving ranking functions
- Output the result
- Documentation and presentation

Libraries and Files

Packages Needed

- Coursera-dl
- Metapy
- Html2text
- Numpy
- Pandas
- Regular Expression

Files

- Original txt and html files
- DataCleaning.py
- Config.toml
- Documents.txt
- CampusWireHeading_Queries.txt
- Qrels.txt
- ModelEvaluation.py
- Search.py

Code Implementation

Data Cleaning

```
 This lecture is the first one about the text clustering. In this lecture, we are going to talk
 This lecture is about the Evaluation of Text Categorization. So we've talked about many differ
 This lecture is about the Latent Aspect Rating Analysis for Opinion Mining and Sentiment Analy
 In this lecture we give an overview of Text Mining and Analytics. First, let's define the term
. This lecture is about the syntagmatic relation discovery, and entropy. In this lecture, we're
This lecture is about the mixture model estimation. In this lecture, we're going to continue dis
Hello welcome to CS410 DSO Text Information Systems. This is an online course offered by Univers
###  **Introduction**  The course project is to give the students hands-on experience on develop
##  Exam Instructions    * A password quiz precedes and unlocks the proctored exam. The proctor
 >> This lecture is about Natural Language of Content Analysis. As you see from this picture, t
 In this lecture, we are going to talk about how to improve the instantiation of the vector spa
 This lecture is about Evaluation of Text Retrieval Systems In the previous lectures, we have t
 This lecture is about the Probabilistic Retrieval Model. In this lecture, we're going to conti
 This lecture is about the feedback in text retrieval. So in this lecture, we will continue wit
 This lecture is about the Learning to Rank. In this lecture, we are going to continue talking
No office hour this week due to Fall Break
In this course, there are two timed exams proctored via [ ProctorU ](https://www.proctoru.com/pc
##  **CS410 Technology Review (4-credit students only)**  **CS410 Technology Review**  The Techr
#  Week 1 Overview  The first six weeks of the course are based on the content of the Text Retri
#  Week 10 Overview  During this week's lessons, you will learn text clustering, including the b
#  Week 11 Overview  During this week's lessons, you will continue learning about various methoc
#  Week 2 Overview  During this week's lessons, you will learn how the vector space model works
#  Week 3 Overview  During this week's lessons, you will learn how to evaluate an information re
#  Week 4 Overview  During this week's lessons, you will learn probabilistic retrieval models ar
#  Week 5 Overview  During this week's lessons, you will learn feedback techniques in informatic
#  Week 6 Overview  During this week's lessons, you will learn how machine learning can be used
#  Week 7 Overview  From Week 7 to Week 12, the lectures are based on the Text Mining and Analyt
```

```python
#Libraries for Data Collection and Cleaning
import os
import html2text
#Preprocess Data
os.remove("documents.txt")
c = open("documents.txt",'a+',encoding = "utf-8")
for filename in os.listdir('Data'):
    if filename.endswith('.txt'):
        with open(os.path.join('Data', filename)) as f:
            content = f.read()
            content = content.replace('\n',' ')
            content = content.replace('[NOISE]',' ')
            content = content.replace('[MUSIC]',' ')
            content = content.replace('[SOUND]',' ')
            content = content.replace('\u2011',' ')
            c.write(content + '\n')

    if filename.endswith('.html'):
        with open(os.path.join('Data', filename), encoding='utf8') as d:
            content = d.read()
            h = html2text.HTML2Text()
            content = h.handle(content)
            content = content.replace('\n',' ')
            content = content.replace('\u2011',' ')
            string = str(content)
            c.write(string + '\n')
c.close()
```

Implementation of Ranking Functions

```python
cfg = "config.toml"
idx = metapy.index.make_inverted_index(cfg)

# testing code - not used in final program
with open(cfg, 'r') as fin:
    cfg_d = pytoml.load(fin)

query_cfg = cfg_d['query-runner']
if query_cfg is None:
    print("query-runner table needed in {}".format(cfg))
    sys.exit(1)

top_k = 10
query_path = query_cfg.get('query-path', 'CampusWireHeading_Queries.txt')
query_start = query_cfg.get('query-id-start', 0)

# get the query from user input
query_text = input("please enter a search query: ")
query = metapy.index.Document()
query.content(query_text.lower())
```

```python
    # get ranking function from user input
    ranking_function = input("""Please select a ranking function. Enter
        1 for Okapi BM25
        2 for Pivoted Length Normalization
        3 for Jelinek-Mercer Smoothing
        4 for Dirichlet Prior Smoothing.
        (Note: If none of these is entered, Okapi BM25 will be used by default.)
""")


    if ranking_function == "2":
        print("Selected Pivoted Length Normalization as ranking function.")
        ranker = metapy.index.PivotedLength(s=0.2)
    elif ranking_function == "3":
        print("Selected Jelinek-Mercer Smoothing as ranking function.")
        ranker = metapy.index.JelinekMercer(.85)
    elif ranking_function == "4":
        print("Selected Dirichlet Prior Smoothing as ranking function.")
        ranker = metapy.index.DirichletPrior(2000)
    else:
        print("Selected Okapi BM25 as ranking function.")
        ranker = metapy.index.OkapiBM25(k1=1.2, b=.75, k3=500)

    top_docs = ranker.score(idx, query, num_results=10)
```

[(54, 1.2830500602722168), (40, 1.2738627195358276), (29, 1.2669488191604614), (63, 1.2639679908752441), (23, 1.2637956142425537), (98, 1.2576667070388794), (5, 1.2554749250411987), (12, 1.254915714263916), (31, 1.2459347248077393), (34, 1.2417280673980713)]

Testing and Fine-tuning Ranking Functions

	Ranker	k1	b	k3	s	lambda	mu	MAP	NDCG
0	Okapi	1.2	0.75	500	NA	NA	NA	0.81	0.85
1	Okapi	1.3	0.8	500	NA	NA	NA	0.79	0.84
2	Okapi	1.1	0.5	500	NA	NA	NA	0.77	0.84
3	Pivoted Length	NA	NA	NA	0.2	NA	NA	0.79	0.83
4	Pivoted Length	NA	NA	NA	0.01	NA	NA	0.79	0.83
5	Pivoted Length	NA	NA	NA	0.5	NA	NA	0.66	0.73
6	JelineK-Mercer	NA	NA	NA	NA	.7	NA	0.66	0.79
7	JelineK-Mercer	NA	NA	NA	NA	.5	NA	0.70	0.81
8	JelineK-Mercer	NA	NA	NA	NA	.85	NA	0.70	0.81
9	Dirichlet Prior	NA	NA	NA	NA	NA	2000	0.66	0.78
10	Dirichlet Prior	NA	NA	NA	NA	NA	1500	0.66	0.78
11	Dirichlet Prior	NA	NA	NA	NA	NA	3500	0.66	0.78

Output the Result

Run - CourseProject

Run: search

```
/usr/local/bin/python3.6 /Users/boyupang/Documents/UIUC/fall21/cs410/CourseProject/search.py
please enter a search query: Probabilistic retrieval models
Please select a ranking function. Enter
        1 for Okapi BM25
        2 for Pivoted Length Normalization
        3 for Jelinek-Mercer Smoothing
        4 for Dirichlet Prior Smoothing.
        (Note: If none of these is entered, Okapi BM25 will be used by default.)
1
Selected Okapi BM25 as ranking function.
Returning top search results:
1. ......lecture is about the Probabilistic Retrieval Model. In this lecture, we're going to continue the discussion of the Text Retrieval Methods. We're going to look at another kind of very different way to design rank

2. ......a overview of text retrieval methods. In the previous lecture, we introduced the problem of text retrieval. We explained that the main problem is the design of ranking function to rank documents for a query. In

3. ......using this kind of probabilistic reasoning where we have made explicit assumptions. And, we know precisely why we have a logarithm here. And, why we have these probabilities here. And, we also have a formula tha

4. ......in depth, including mixture models and how they work, Expectation-Maximization (EM) algorithm and how it can be used to estimate parameters of a mixture model, the basic topic model, Probabilistic Latent Semanti

5. ......about a number of retrieval methods. We start with the overview of vector space model and the probabilistic model and then we talked about the vector space model in depth. We also later talked about the language

6. ......is about query likelihood, probabilistic retrieval model. In this lecture, we continue the discussion of probabilistic retrieval model. In particular, we're going to talk about the query light holder retrieval f

7. ......Text Mining called Contextual Probabilistic Latent Semantic Analysis. In this lecture, we're going to continue discussing Contextual Text Mining. And we're going to introduce Contextual Probablitistic Latent Sem

8. ......course covers both text retrieval and text mining, so as to provide you with the opportunity to see the complete spectrum of techniques used in building an intelligent text information system. Building on two MO

9. ......lessons, you will learn probabilistic retrieval models and statistical language models, particularly the detail of the query likelihood retrieval function with two specific smoothing methods, and how the query l

10. ......smoothing methods for language models used in probabilistic retrieval model. In this lecture, we will continue the discussion of language models for information retrieval, particularly the query likelihood retr


Process finished with exit code 0
```

Live Test